

CAPTURE IMAGE DENGAN PENANDA JARI

Rendra Maulana P., Rizky Yuniar Hakkun, S.Kom, M.T; Setiawardhana, S.T, M.T
Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya
Email: renatzz_itce08@yahoo.com

Abstrak

Sejauh ini kita telah banyak mengenal penggunaan kamera, baik yang digunakan untuk komersial maupun non-komersial di masyarakat. Menanamkan kecanggihan untuk pengambilan suatu gambar (capture image) bertujuan untuk memfasilitasi pengguna. Tujuan dari proyek akhir ini adalah program dapat melacak kedua object warna yaitu biru dan merah sehingga membuat aplikasi ini dapat mengcapture adanya sebuah gambar yang kita inginkan. Pengolahan gambar untuk melacak object menggunakan sumber gambar pengolahan perpustakaan terbuka yaitu Aforge.Net dan ditulis dalam Visual C# 2010. Posisi hasil pelacakan objek dapat digunakan sebagai masukan untuk menemukan masing-masing warna objek dan menentukan koordinat posisi warna tersebut, hingga membuat jarak antara masing-masing kedua warna dapat diketahui sehingga kita mampu mengambil gambar secara mandiri.

Kata kunci : *Pengambilan Gambar, Pelacakan Warna, Computer Vision, Aforge.Net*

Abstract

1 Pendahuluan

1.1 Latar Belakang

Pada penelitian sebelumnya telah dilakukan pengambilan gambar dengan menggunakan *marker*. Pengguna dapat mengambil gambar secara visual sesuai posisi yang diinginkan. Penelitian ini masih dapat dikembangkan lagi untuk meningkatkan efektifitas kerja sistem salah satunya adalah kemampuan untuk mengambil gambar dengan penanda sebuah jari.

1.2 Tujuan

Tujuan dari penelitian tugas akhir ini adalah aplikasi mampu mendeteksi adanya warna, khususnya merah dan biru. Sehingga dari warna tersebut mampu menghasilkan suatu gambar sesuai dimana posisi yang kita ambil.

1.3 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut :

- Pengenalan obyek target berdasarkan pada warna obyek dalam citra.
- Model obyek adalah warna merah dan biru dengan ukuran minimal pada pada image kamera 25 x 25 pixel.

- Background atau dinding ruangan yang digunakan berbeda dengan warna obyek target.
- Proses pengambilan gambar sejajar sumbu x (horizontal).
- Pengambilan gambar terjadi saat jarak antar masing-masing warna kurang lebih sama dengan (\leq) 50 pixel.

2 Teori Penunjang

2.1 Color Filtering

Color filtering merupakan *filter* yang digunakan untuk memfilter warna yang diinginkan pada gambar atau citra berdasarkan nilai komponen pada RGB. Kombinasi RGB inilah yang kemudian dijadikan *filter* sebagai penentu sebuah warna diloloskan atau tidak.

2.2 Grayscale Filter

Grayscale filter digunakan untuk mengubah gambar berwarna menjadi gambar hitam putih, dengan cara mengubah efek warna dari masing-masing pixel menjadi derajat keabuan.

2.3 Blob Counter

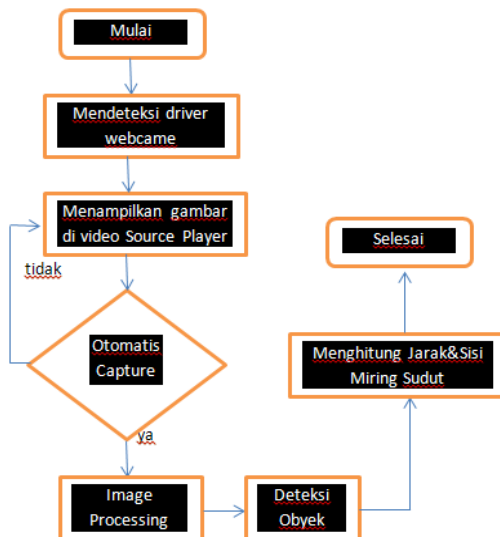
Blob counter digunakan untuk menghitung obyek, memfilter obyek, mengekstrak obyek dan mendapatkan dimensi obyek yang ada digambar, yang dipisahkan oleh latar belakang hitam dengan menggunakan prinsip algoritma pelabelan komponen terhubung. Algoritma ini memperlakukan semua pixel dengan nilai kurang atau sama dengan batas ambang *background* sebagai latar belakang, sedangkan pixel dengan nilai yang lebih maka diperlakukan sebagai pixel obyek

2.4 Aforge.Net

AForge.NET merupakan *framework open source C#* yang dirancang bagi para pengembang dan peneliti di bidang komputer vision dan *artificial intelligence* yang meliputi pengolahan citra, jaringan syaraf tiruan, algoritma genetika, logika fuzzy, *machine learning*, dan robotika.

3. Perancangan Sistem

3.1 Perancangan Perangkat Lunak



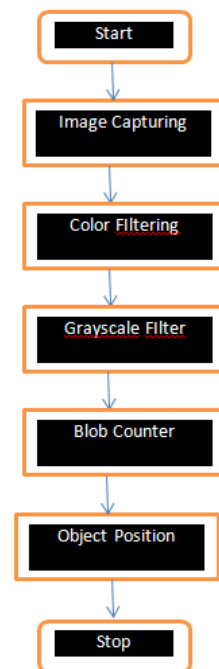
Gambar 3.1 Flowchart Sistem Program.

3.2 Perancangan Pengolahan Citra

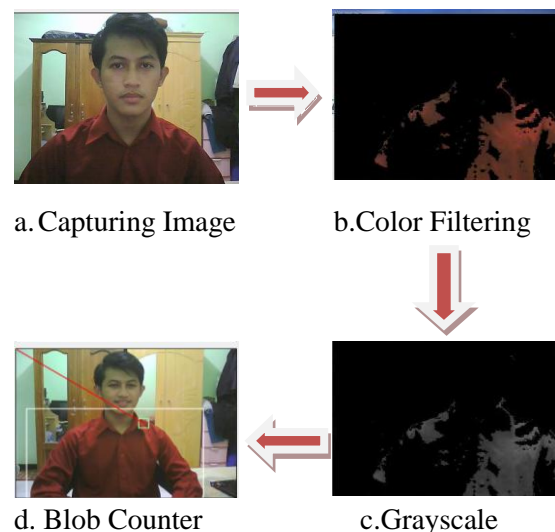
Pada tugas akhir ini, untuk proses pengolahan citra menggunakan *library AForge.Imaging*, sedangkan untuk

proses pengolahan video menggunakan *library AForge.Video*. Pengolahan video digunakan untuk mendeteksi *driver webcam* dan menampilkan gambar di *videoSourcePlayer* sehingga dapat digunakan untuk proses pengolahan citra.

Proses pengolahan citra menggunakan *library AForge.Imaging* yang memiliki berbagai macam *class* dan *class* yang dibutuhkan untuk penjejak objek yaitu *class colorFiltering*, *class grayscale* dan *class blobCounter*.



Gambar 3.2 Flowchart Pelacakan Obyek



Gambar 3.3 Proses Pengolahan citra Pelacakan Obyek

a. Capturing Image

Proses capturing image menggunakan webcam komic seri WEC201. Resolusi webcam yang digunakan yaitu 1,3 mp. Untuk pendeteksian webcam digunakan *library Aforge.NET* yaitu *library Aforge.video*. Selain digunakan untuk mendeteksi driver webcam juga digunakan untuk menampilkan citra di komponen *videoSourcePlayer* sehingga dapat digunakan untuk pengolahan citra.

b. Color Filtering

Citra hasil capturing image kemudian difilter menggunakan *color filtering*. *Color Filtering* merupakan proses untuk memfilter warna yang diinginkan pada suatu gambar berdasarkan komponen RGB. Penetapan nilai batas pada filter dapat dilakukan dengan mengubah ukuran batas nilai pada ruang warna RGB. Misal warna yang diinginkan warna merah dengan komponen nilai RGB $100 < R < 255, 0 < G < 70, 0 < B < 70$ dari batas rentang nilai tersebut maka warna yang memiliki nilai paling tinggi itulah yang akan muncul atau dapat dikatakan sebagai obyek sedangkan warna yang lain dianggap *background* atau dalam hal ini menjadi hitam.

c. Grayscale Filter

Citra hasil *color filtering* kemudian dikonversi menjadi *grayscale* agar dapat digunakan oleh *class BlobCounter*. Proses konversi dari citra berwarna menjadi grayscale menggunakan koefisien dari ITU-Recommendation BT.709. $\text{Grayscale} = 0.2125 * \text{red} + 0.7154 * \text{green} + 0.0721 * \text{blue}$.

d. BlobCounter

Obyek citra hasil *grayscale* kemudian diidentifikasi dengan menggunakan *blobcounter*. Identifikasi obyek ini digunakan untuk menghitung *blob*, memfilter *blob*, mengekstrak *blob*, dan mendapatkan dimensi *blob*. Di dalam *class blobCounter* terdapat metode *GetObjectRectangle* yang digunakan untuk mendapatkan bentuk kotak dari obyek.

Setelah bentuk otak didapat maka proses selanjutnya yaitu proses penggambaran kotak pada obyek yang terdeteksi menggunakan *class graphics* dan *class pen*. Dalam perancangannya obyek mempunyai luas minimal 100 piksel. Jika dalam *image* kamera terdapat obyek jamak maka *blobCounter* akan memprioritaskan luas obyek yang lebih dominan yang akan dideteksi terlebih dahulu.

e. Object Position

Citra hasil *blobCounter* dapat digunakan untuk menghitung posisi obyek dengan memanfaatkan metode yang ada pada *class blobCounter* yaitu *GetObjectRectangle*. Metode ini digunakan untuk mendapatkan bentuk kotak obyek sehingga dapat dicari jarak obyek pada sumbu x dari tepi kiri dan jarak sumbu y dari tepi atas serta lebar dan tinggi obyek.



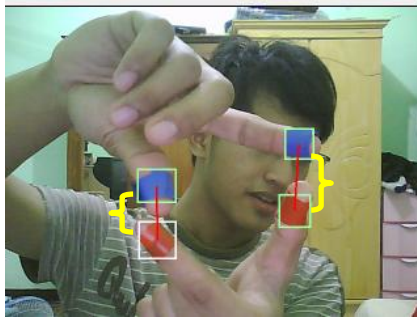
Gambar 3.4 Perhitungan Posisi Obyek

Posisi obyek pada koordinat x yaitu jarak objek sumbu x dari tepi kiri dijumlahkan dengan setengah dari lebar objek.

```
positionObjectX=objectRect.X+objectRect.Width/2
```

Posisi obyek pada koordinat y yaitu jarak objek sumbu y dari tepi atas dijumlahkan dengan setengah dari tinggi objek.

```
positionObjectY=objectRect.Y+objectRect.Height/2
```

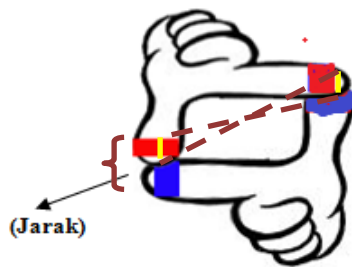


Gambar 3.4 Perhitungan Jarak Capture Image

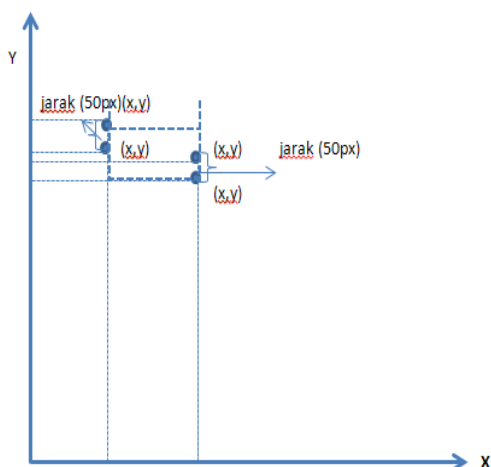
Pengambilan gambar terjadi saat obyek image merah dan biru , memenuhi kondisi sebagai berikut :

```
if (jarak1 < 50 && jarak2 < 50)
{
.....
}
```

dimana jarak adalah sebuah variable yang didapat dengan mengkondisikan jarak 1 dan jarak 2 apabila kurang dari 50 pixel maka proses pengambilan gambar dapat terwujud.

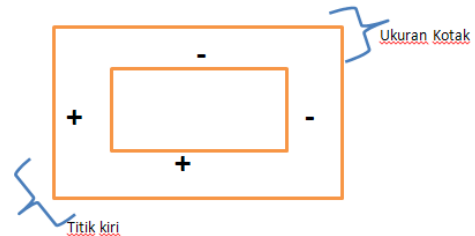


Gambar 3.5 Peletakan Obyek Warna



Gambar 3.6 Pembentukan Hasil Capture dengan Sistem Koordinat Kartesius

Sebuah hasil pembentukan gambar hasil *crop* (memotong gambar) atau dengan kata lain dalam penelitian ini adalah *capture image*. Maka dijelaskan bahwa dimana kondisi dari setiap posisi titik warna akan direpresentasikan pada sistem koordinat kartesius

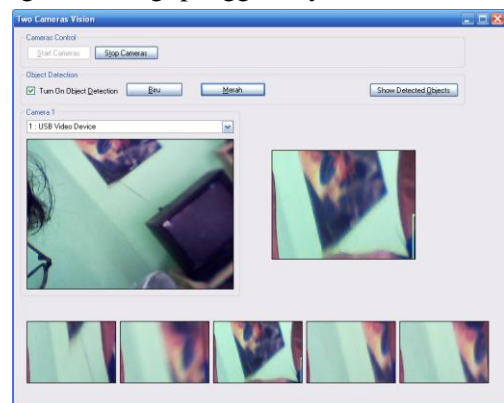


Gambar 3.7 Sistem Penambahan dan Pengurangan Piksel

Dari pembentukan gambar yang dihasilkan ,hasil suatu image perlu adanya penambahan pada bagian sektor titik kiri dan pengurangan pada bagian sektor ukuran kotak. Hal ini dikarenakan pada saat pengambilan gambar, tangan posisi kita mengambil gambar terdapat juga pada hasil pengambilan gambar. Maka perlu adanya sistem tersebut, pada gambar diatas (Gb.3.7).

4. Hasil dan Analisa

Pengujian dilakukan untuk memperoleh parameter penting sebagai acuan untuk pengembangan selanjutnya, jika masih diperlukan suatu penyempurnaan. Pengujian dengan menentukan wilayah pada saat pengambilan gambar , akan memiliki nilai penting karena pengambilan wilayah yang sesuai kita harapkan akan mempunyai citarasa yang sesuai bagi penggunaanya



Gambar 4.1 Hasil Deteksi Warna

Analisa :

Dari hasil percobaan diatas menunjukan bahwa posisi kamera sejajar dengan arah kita memandang. Maka akan menghasilkan suatu image yang berada di depan kita bila pada saat itu kita sedang ingin melakukan pengambilan gambar. Hasil gambar yang ada diatas posisi gambar tampak miring hal ini dikarenakan penempatan letak posisi kamera tidak stabil dan pengaruh jenis kamera sangat mempengaruhi hasil kualitas gambar.

4.3 PENGUJIAN DAN ANALISA HASIL

(Keseluruhan)

Bagian ini akan menyajikan beberapa pengujian sistem secara keseluruhan. Sistem secara keseluruhan sendiri meliputi deteksi warna dan menampilkan daerah garis kotak pada warna serta interaksi diantaranya.

Pengujian ini dilakukan secara bertahap dan berhubungan satu dengan yang lain. Apabila tahap pertama belum berhasil maka pengujian berikutnya akan tertunda sampai pengujian tahap pertama selesai dilakukan, maka baik dari segi warna maupun sistem harus siap dan dilakukan bertahap sehingga semua berjalan dengan baik serta dapat dianalisa kekurangan yang ada.

Pada saat pengujian sistem keseluruhan dapat mengambil gambar objek, dimana objeknya ditentukan sesuai keinginan. Objek ini akan diletakkan pada posisi yang berbeda-beda terhadap kamera.

Tabel 4.1 Pengujian Gambar Terhadap Pengaruh Cahaya Indoor

NO	DAERAH CAPTURE	CAHAYA	HASIL
1	Kanan Atas	Cerah	Berhasil
2	Kanan Atas	Tidak Cerah	Berhasil
3	Atas	Cerah	Berhasil
4	Atas	Tidak Cerah	Berhasil
5	Kiri Atas	Cerah	Berhasil
6	Kiri Atas	Tidak Cerah	Berhasil
7	Kiri	Cerah	Berhasil
8	Kiri	Tidak Cerah	Berhasil

9	Kanan	Cerah	Berhasil
10	Kanan	Tidak Cerah	Berhasil
11	Kanan Bawah	Cerah	Berhasil
12	Kanan Bawah	Tidak Cerah	Berhasil
13	Bawah	Cerah	Berhasil
14	Bawah	Tidak Cerah	Berhasil
15	Kiri Bawah	Cerah	Berhasil
16	Kiri Bawah	Tidak Cerah	Berhasil
17	Tengah	Cerah	Berhasil
18	Tengah	Tidak Cerah	Berhasil

Analisa :

Pengujian diatas dilakukan untuk menguji kemampuan sistem dalam pencahayaan. Dalam proses pengambilan gambar diperlukan deteksi warna yang sangat sensitive terhadap pencahayaan. Pada table diatas terdapat kolom cahaya yang menentukan terdeteksinya warna.

Cahaya cerah diperoleh pada ruangan ukuran 5 x 5 dengan cahaya lampu 16 watt maka dari setiap pengambilan gambar masi mampu ditangkap dengan baik, sedangkan pada cahaya tidak terang dilakukan pada saat pagi hari pukul 08.00, dengan ventilasi yang tidak terlalu besar (satu jendela ukiuran standart) . Hasil dari pengambilan gambar ternyata memiliki pengaruh pada saat pengambilan gambar yakni dimana warna sebagai media penanda untuk mengambil gambar disaat cahaya berkurang maka nilai RGB dari warna tersebut pun berubah, pengambilan dapat dilakukan dengan baik dengan cara mengubah nilai RGB pada button *TunerObjectFilter* .

Tabel 4.2 Pengujian Gambar Terhadap Pengaruh Cahaya Outdoor

WILAYAH	CAHAYA	WAKTU	HASIL
Kanan Atas	Cerah	08.00-09.00	Berhasil
Kanan	Tidak	17.00-	Berhasil

Atas	Cerah	18.00	
Atas	Cerah	08.00-09.00	Berhasil
Atas	Tidak Cerah	17.00-18.00	Tidak Berhasil
Kiri Atas	Cerah	08.00-09.00	Berhasil
Kiri Atas	Tidak Cerah	17.00-18.00	Berhasil
Kiri	Cerah	08.00-09.00	Berhasil
Kiri	Tidak Cerah	17.00-18.00	Berhasil
Kanan	Cerah	08.00-09.00	Berhasil
Kanan	Tidak Cerah	17.00-18.00	Berhasil
Kanan Bawah	Cerah	08.00-09.00	Berhasil
Kanan Bawah	Tidak Cerah	17.00-18.00	Tidak Berhasil
Bawah	Cerah	08.00-09.00	Berhasil
Bawah	Tidak Cerah	17.00-18.00	Berhasil
Kiri Bawah	Cerah	08.00-09.00	Berhasil
Kiri Bawah	Tidak Cerah	17.00-18.00	Tidak Berhasil
Tengah	Cerah	08.00-09.00	Berhasil
Tengah	Tidak Cerah	17.00-18.00	Berhasil

Analisa :

Cahaya cerah diperoleh di luar ruangan wilayah sekitar rumah, pengambilan gambar dilakukan pada saat pagi hari pukul 08.00-09.00, Hasil dari percobaan tersebut dimana pada waktu itu disaat cahaya tiba-tiba terik dengan intensitas yang tinggi maka pendeteksian pun mengalami gangguan yang lumayan besar, sistem menjadi kacau hal ini dikarenakan nilai RGB dari yang telah ditentukan tidak stabil dengan intensitas

cahaya luar yang beubah-ubah, sedangkan pada cahaya tidakcerah yang dilakukan pada waktu 17.00-18.00 intensitas cahaya cukup stabil namun adakalanya deteksi warna tidak dapat dilakukan dengan baik hal ini dikarenakan intensitas cahaya yang minim membuat nilai RGB warna obyek menjadi menurun yaitu lebih gelap.

Pengujian selanjutnya adalah pengujian pengambilan gambar terhadap jarak marker dari kamera. Jarak target dilakukan bervariasi mulai 30cm sampai 1,5 meter. Pengambilan gambar dilakukan sebanyak 10kali dalam setiap jarak tertentu.

Tabel 4.3 Pengujian Pengambilan Gambar Terhadap Kamera dan Warna

JARAK	BER HASIL	GAGAL	PRESEN TASE
10	5	0	100%
20	5	0	100%
30	5	0	100%
40	4	1	85%
50	4	1	85%
60	0	5	0%
70	0	5	0%

Analisa :

Dari hasil percobaan diatas menunjukan pengambilan gambar tidak mampu dilakukan dengan jarak yang jauh (<40 cm). hal ini dikarenakan pertama adalah faktor kamera yang sangat menentukan , kedua jarak pembukuan *blob* hantya mampu mengenal bentuk obyek minimal 75 x 75 pixel. Dari hal ini lah jarak mempengaruhi ddalam proses pengamblan gambar dengan menggunakan penanda jari.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisa yang telah di bahas pada bab sebelumnya maka dapat diberikan beberapa kesimpulan sebagai berikut :

- 1) Kamera berpengaruh besar terhadap pengambilan suatu gambar
- 2) Banyaknya jenis warna memiliki sifat pemendaran yang berbeda-beda
- 3) Kondisi cahaya merupakan factor penentu dalam pengambilan suatu gambar, intensitas yang terlalu tinggi ataupun terlalu rendah tidak begitu baik untuk pendeteksian warna

DAFTAR PUSTAKA

- [1]. Basuki, Ahmad, *Pengolahan Citra*, Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya Institut Teknologi Surabaya, Surabaya.
- [2]. <http://www.aforgenet.com/framework/> (diakses 10 april 2011).
- [3]. Hartanto,Budi, *Membuat Program-Program Keren dengan Visual C#.Net Secara Mudah*, ANDI, Yogyakarta, 2009.
- [4]. Alamsyah, *Pengolahan Citra Digital Untuk Sistem Penjejak Sasaran Pada Model Roket LAPAN*, Skripsi S-1, Institut Teknologi Bandung, Bandung, 2007.
- [5]. Nahla,G,S, *Tracking Bola Menggunakan Robotino*, Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya Institut Teknologi Surabaya, Surabaya.
- [6]. Kurnia, Rahmadi, *Penjejak Target Benda Pada Gerakan Linier Berdasarkan Warna*, SNATI ,Yogyakarta,2009.
- [7]. Zhong D and chang S. F *Video Object Model and Segmentation for Content-Based Video Segementation* in IEEE International Conference of Circuits and System,Hong Kong, 1997.
- [8]. Hapsari,, Widi, *Segementasi Warna Citra dengan Deteksi WarnaHSV untuk mendeteksi Obyek*, Yogyakarta,2010.
- [9]. <http://cosaviora.blogspot.com/2010/11/computer-vision.html> (diakses 10 april 2011).
- [10]. http://en.wikipedia.org/wiki/Computer_vision (diakses 1 Mei 2011).
- [11]. <http://www2.eepis-its.edu/~riyanto/citra-bab2.pdf>
- [12]. <http://planetmath.org/encyclopedia/EuclideanDistance.html> (diakses 2 Mei 2011).
- [13]. Addison, Wesley, *Follows the Format That Made Effective C#, Inggris*, 1998.
- [14]. http://www.aforgenet.com/articles/lego_pan_tilt_camera/ (diakses 15 april 2011).